

Desain Class Dan Analisis Operator Pada Class Integer Tak Terbatas

Bandung Arry Sanjoyo

Jurusan Matematika FMIPA - ITS

email:bandung@matematika.its.ac.id

Ringkasan

Pada makalah ini dibahas tentang representasi bilangan bulat (integer) dengan ukuran digit yang sangat besar. Representasi data dan operasi yang terkait diwujudkan dalam bentuk class integer. Struktur data yang digunakan untuk menyimpan bilangan berupa list (array) of character. Sedangkan operator-operator terkait yang bekerja pada integer adalah perbandingan, penjumlahan, pengurangan, perkalian dan pembagian. Analisis operator dibahas secara analitis.

Keyword: *Class, integer, worstcase.*

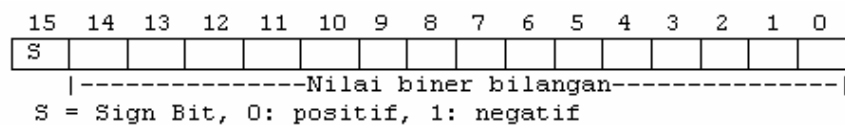
1. Pendahuluan

Dalam sistem bilangan, bilangan integer (bulat) 0, +1, +2, banyak dipakai dalam berbagai bidang termasuk dalam bidang sains, teknik, sosial, ataupun humaniora. Implementasi bilangan integer dalam komputer terbatas pada berapa besar bit yang dialokasikan untuk integer tersebut. Berbagai macam bahasa pemrograman yang ada saat ini, seperti Visual C++, Visual Basic, Matlab dan lainnya, memberikan tempat memori untuk menyimpan integer sebesar 64bit. Oleh karena itu bilangan integer terbesar yang dapat disimpan tidak dapat melebihi $2^{63} - 1$, untuk di Matlab maksimal $2^{53} - 1$ atau kalau dalam digit decimal tidak lebih dari 16 digit[3].

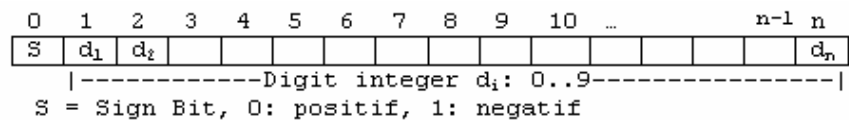
Permasalahan sekuritas data (encryption, decryption) akan melibatkan bilangan prima (bagian dari integer) dengan jumlah digit yang sangat besar. Semakin besar bilangan prima yang dipilih, keamanan datanya lebih terjamin. Keterlibatan integer ini selalu diikuti dengan kebutuhan akan operator seperti penjumlahan, pengurangan, perkalian, dan pembagian, atau bahkan lebih dari operator yang disebutkan tadi. Oleh karena itu, dalam makalah ini akan disajikan bagaimana merepresentasikan integer yang sangat besar (tak terbatas) dan bagaimana mendefinisikan proses operasi pada integer tersebut.

2. Representasi Integer

Himpunan $\dots, -2, -1, 0, 1, 2, 3, \dots$ merupakan himpunan bulangan bulat (integer) dan biasa dinotasikan dengan \mathbb{Z} . Sehingga nilai integer bisa mendekati tak berhingga atau banyaknya digit pada integer tersebut sangat besar. Dalam representasi di memori komputer, integer dialokasikan dalam berbagai kapasitas tempat memori (1 byte hingga 8 byte). Integer 2 byte disimpan dalam format sebagai berikut[3].



Bilangan terkecil dan terbesar yang dapat disimpan dalam integer 2 byte adalah -32,768 sampai dengan 32,767. Untuk kapasitas integer 64 bit atau 8 byte, nilai integer dapat ditampung adalah 9,223,372,036,854,775,808 sampai dengan 9,223,372,036,854,775,807. Oleh karena itu untuk integer yang lebih dari 19 digit (integer tak terbatas) tidak mampu ditampung. Penyimpanan integer tak terbatas disimpan dalam bentuk sederetan karakter yang dinamis atau *array of character*.

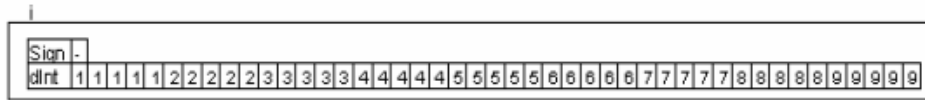


Bentuk representasi di atas dapat dinyatakan dalam bentuk struktur data berikut ini.

```
class intBig {
private:
char    Sign;
char*   dInt;
public: /* metoda, operator */
}
```

Integer i=-111112222233333444445555566666777778888899999 direpresentasikan

dalam bentuk berikut ini.



Dengan struktur data seperti di atas, integer dengan banyak digit sembarang akan bias ditampung. Namun demikian terbatas juga pada besarnya free memory yang tersedia.

3. Pembentukan Class Integer Tak Terbatas

Suatu hal yang mendasar dalam desain class adalah identifikasi objek (benda) dan kerja/operasi yang bekerja pada objek tersebut. Pada integer dikenali:

- i. Objek: tanda bilangan (+/-) atau disimbolkan dengan Sign, dan
- ii. Operasi:
 - (a) Pendefinisian objek atau constructor.
 - (b) Operasi perbandingan: untuk membandingkan dua integer besar.
 - (c) Operasi penjumlahan: untuk menjumlahkan dua buah integer besar.
 - (d) Operasi pengurangan: untuk mengurangi dua buah integer besar.
 - (e) Operasi perkalian: untuk mengalikan dua buah integer besar.
 - (f) Operasi pembagian: untuk membagi dua buah integer besar.
 - (g) Operasi menampilkan integer besar.

Operasi yang selain disebutkan di atas belum dibahas dalam penelitian ini. Identifikasi di atas dapat dinyatakan dalam suatu bentuk desain class integer besar (intBig) sebagai berikut.

```
class intBig {
private:
    char Sign; /* unt simpan tanda + atau - */
    char* dInt; /* unt simpan sederet digit int */
public:
    int operator<(intBig n);
    int operator<=(intBig n);
    int operator>(intBig n);
    int operator>=(intBig n);
    int operator==(intBig n);
    intBig operator+(intBig n);
    intBig operator-(intBig n);
    intBig operator*(intBig n);
    intBig operator/(intBig n);
    display(intBig n); };

```

4. Desain Dan Analisis Algoritma Untuk Operator Pada Class Integer Tak Terbatas

Strategi atau metoda dari operasi perbandingan, pengurangan dan penjumlahan dapat dilihat pada Bandung A.S. [1]. Dalam penelitian ini akan dibuat algoritma untuk perkalian dan pembagian. Algoritma Perkalian Ada beberapa cara mengalikan dua buah integer, diantaranya adalah penjumlahan berulang dan metoda geser kiri.

Penjumlahan berulang Perkalian dua integer $a \times b$, dapat dilakukan dengan cara sebagai berikut. $a \times b = b + b + \dots + b$, penjumlahan b dengan b dikumulasikan sebanyak a kali. Strategi demikian dapat diimplementasikan sebagai berikut.

Input : integer besar a dan b .

Output : integer besar r .

Algoritma :

```
perkalian(a,b)
  r=0;
  for i=1 to a do
    r=r+b;
  end for;
end of perkalian;
```

Dalam worst case, algoritma diatas bekerja sebanyak a buah penjumlahan integer besar. Untuk kasus dimana nilai a jauh lebih besar dari pada panjang s , dilakukan perbaikan algoritma sebagai berikut.

Metoda Geser Kiri ini sudah dikenal sejak sekolah dasar, dan digambar dengan contoh berikut ini. 321×1234 dilakukan dengan cara:

```

      321
    1234
----- *
      1284
     963*
    642**
   321***
----- +
  396114
```

Pada contoh diatas $a * b$, integer yang dikalikan $a = 321$ dan pengali $b = 1234$. Integer a dikalikan dengan setiap digit pada integer b , dari digit paling kanan ke kiri. Pada setiap perkalian dilakukan pergeseran satu digit ke kiri. Dan

selanjutnya dilakukan penjumlahan terhadap hasil perkalian dari setiap digit n . Strategi perkalian di atas dapat dikodekan sebagai berikut.

Input : integer besar a dan b , dengan $m = |a|$ dan $n = |b|$

Output : integer besar r .

Algoritma :

```
Perkalian(a,b,m,n)
  r = 0;
  for i = 1 to m
    d = ai;
    temp = 0;
    for j = 1 to d
      temp = temp + b;
    end;
    r = r + temp;
    r=concat(r,'0');
end; end of perkalian;
```

Dalam worst case, algoritma diatas bekerja $10m$ buah operasi jumlah dan concat. Untuk kasus dimana nilai $m < n$ akan bekerja lebih cepat dari pada $m > n$.

Algoritma Pembagian Metoda Pengurangan Berulang Pembagian dua integer a/b , dapat dilakukan dengan cara sebagai berikut: $a/b = a - b - b - \dots - b$, pengurangan a dengan b sebanyak n kali, n adalah hasil pembagian. Strategi demikian dapat diimplementasikan sebagai berikut.

Input : integer besar s dan d , dengan $s > d > 0$

Output : integer besar m .

Algoritma :

```
pembagian(s,d)
  r=s; m=0;
  while r < d do
    r=r-d;
    m=m+1;
  end while
end of pembagian;
```

Dalam worst case, algoritma diatas bekerja s/d . Untuk kasus dimana nilai s/d jauh lebih besar dari pada panjang s , dilakukan perbaikan algoritma sebagai berikut.

Metoda Paragapit Cara pembagian paragapit s/d digambarkan dengan contoh berikut ini.

<div> 15 121445 8 0 9 6 120 <hr/> 14 0 <hr/> 144 135 <hr/> 96 90 <hr/> 5 </div>	d=15; s=121445;					
	i	temp	pt	r	J	hasil
	-	12	2	-	-	
	-	121	3			
	3			121	0	
				106..1	1..8	
		14				8
	4			14	0	
		144				80
	5			144	0	
				129..9	1..9	
		96				809
	6			96	0	
				85..5	1..6	8096
	Sisa		5	Hasil		8096

Strategi pembagian di atas dikodekan sebagai berikut.

Input : integer besar s dan d , dengan $s > d > 0$, $m = |s|$ dan $n = |d|$

Output : integer besar hasil

Algoritma :

```

pembagian(s,d,m,n)
  temp=s(1:n];
  pt=n;
  hasil='';
  if temp < d then
    temp=cancate(temp,s(pt+1));
    pt=pt+1;
  for i=pt to m do
    r=temp; j=0;
    while r < d do
      r=r-d; j++;
    hasil=concate(hasil,j);
    if i < n then
      temp=concate(r,s[i+1]);
    end for
  end of pembagian;

```

Dalam worst case, algoritma diatas bekerja kurang dari $mn - n^2 + n = \mathcal{O}(mn)$.

5. Kesimpulan

- (a) Integer yang dinyatakan dalam class berbentuk sederetan karakter 0–9 memungkinkan untuk menyimpan integer dengan panjang digit tak terbatas sesuai kemampuan memori computer.
- (b) Operator perkalian integer dengan integer dapat dilakukan dengan kecepatan proses $10\times$ banyaknya digit pada integer yang dioperasikan.
- (c) Operator pembagian integer dengan integer dapat dilakukan dengan kecepatan proses tidak lebih dari $mn - n^2 + n$.

Pustaka

- [1] Bandung Arry S, 2004, *Handout Kuliah Matematika Komputasi*, Pasca Sarjana Program Studi Matematika.
- [2] Kamin, 1996, S.N. dan Reinghold, E.M., *Programming With Class - A C++ Introduction to Computer Science*, The McGraw-Hill Companies, Inc., International Edition.
- [3] Brian Brown, 1984-2000, *Data Structures And Number Systems*, <http://www.ibilce.unesp.br>.
- [4] ———, 2001-2003, *The Long Division Algorithm*, <http://www.mathpath.org/Algor>.